

Brain MRI Patient Identification Based on Capsule Network

Shuqiao Liu, Junliang Li and Xiaojie Li*

Chengdu University of Information Technology, Chengdu, 610225, China

*Corresponding Author: Xiaojie Li. Email: lixj@cuit.edu.cn

Received: 20 May 2020; Accepted: 15 August 2020

Abstract: In the deep learning field, “Capsule” structure aims to overcome the shortcomings of traditional Convolutional Neural Networks (CNN) which are difficult to mine the relationship between sibling features. Capsule Net (CapsNet) is a new type of classification network structure with “Capsule” as network elements. It uses the “Squashing” algorithm as an activation function and Dynamic Routing as a network optimization method to achieve better classification performance. The main problem of the Brain Magnetic Resonance Imaging (Brain MRI) recognition algorithm is that the difference between Alzheimer’s disease (AD) image, the Mild Cognitive Impairment (MCI) image, and the normal image is not significant. It is difficult to achieve excellent results using a multi-layer CNN. However, CapsNet can be in the case of a shallower network, which can accommodate more useful feature information for identifying brain MRI. In this paper, we designed a shallow CapsNet to identify patients with brain MRI by binary classification. Compared with VGG16, Resnet34, DenseNet121 and ResNeXt50. Experimental results illustrate that CapsNet is superior to CNN network in its accuracy and F1-score. The indicators were 86.67% and 83.33%, respectively. Furthermore, we show that the capsule network shows excellent performance in brain MRI recognition compared with those popular networks.

Keywords: Capsule Net; dynamic routing; CNN; Alzheimer’s disease

1 Introduction

Alzheimer’s disease (AD) is a degenerative neurological degenerative disease that develops insidiously. The patient’s ability to perform daily life is progressively reduced with a variety of neuropsychiatric and behavioral disorders. Frequent in the elderly, usually the condition is progressively aggravated, and gradually loses the ability to live independently, 10 to 20 years after the onset of death due to complications. The pre-clinical phase of Alzheimer’s disease is also known as mild cognitive impairment (MCI), a transitional state between normal and severe. Therefore, the accurate diagnosis of Alzheimer’s disease and mild cognitive impairment is of great significance.

Brain magnetic resonance imaging is an important imaging diagnostic tool for brain diseases. Disease identification and prediction based on MRI images is an important issue in the medical field. Traditional medical imaging diagnostic methods rely on years of experience and clinical research by clinicians to manually identify patients with corresponding diseases. The emergence of machine learning methods provides an intelligent solution to the identification problem in the medical field.

Traditional medical imaging diagnostic methods rely on years of experience and clinical research by clinicians to manually identify patients with corresponding diseases. The emergence of machine learning methods provides an intelligent solution to the identification problem in the medical field. Among them, the deep learning method has been widely used in various research fields. Classification and segmentation networks such as ResNet [1–3] exhibit excellent performance in various neighborhoods of computer vision. These networks are based on various variants developed on the Convolutional Neural Network



(CNN). At the same time, this kind of method has been successfully applied to the identification and prediction of medical imaging research fields such as Dilated Heart Disease and prostate cancer [4–5].

Capsule networks (CapsNet) represent a recent breakthrough in neural network architectures [6]. Because the difference between brain MRI images and normal human images is not significant, multi-layer CNN networks are difficult to achieve better results, CapsNet can accommodate more feature information in the case of shallower networks. And get a better classification effect. Based on the literature [6], this paper firstly used a shallow capsule network to identify three-class patients with brain 3D MRI images to test the recognition performance of the capsule network structure and such medical data. In this paper, the method is used in the brain 3D MRI dataset, and compared with ResNet18 [1], ResNeXt50 [7], DenseNet121 [8] and VGG16 [2]. Experimental results illustrate that CapsNet is superior to CNN network in accuracy and F1-score. Compared with traditional methods and multi-layer CNN networks, the capsule network has excellent performance in brain MRI recognition, which can effectively identify whether brain MRI images are patients, and indicates that capsule neurons act as a network. Structural unit performance is better than traditional CNN [6].

2 Related Work

The network structure based on CNN has achieved great success in the fields of computer vision and medicine [9–12]. Compared with traditional medical imaging, artificial design or designation of image features for patient diagnosis, machine learning, especially deep learning methods, can assist physicians to provide preliminary quantitative and qualitative evaluation of medical diagnosis, thereby saving a lot of labor costs. However, the difference between the image of the patients' brain MRI and the healthy people's brain MRI is not significant (as shown in Fig. 1. a: The brain MRI of an Alzheimer's disease patient; b: a mild cognitive impairment patient; c: a healthy person). The shallower classification methods such as multi-layer CNN and SVM are difficult to extract effective image classification features from the weaker images. Therefore, it is not possible to achieve a good recognition effect.



Figure 1: Brain magnetic resonance imaging

CNN is good at detecting specific features in pictures, such as detecting nose and eyes, but it is difficult to find out the relationship between features, such as the size and direction of the view. A face photo exchanged with the nose and nose may be misidentified by CNN as a real face. Based on the visualization system theory proposed by Hinton, Capsule and Capsule Net (CapsNet) came into being to overcome the shortcomings of the CNN method [6]. Hinton pointed out that there is a tree-like analytical structure for each fixed visual position, and each parse tree consists of a fixed multi-layer the neural network [13]. Each layer of neural network is composed of many different “capsule” neurons [14]. Unlike traditional neuron output scalar values, capsule neurons are vectors that contain some special information, such as similarity, direction, size, angle, etc., and finally, determine the activation value of the capsule neurons based on the modulus of the vector (output value). CapsNet uses the squashing algorithm as the vector activation function and uses the dynamic routing algorithm to replace the dropout of the fully connected network in the CNN, making the network features more multi-layered and achieving better algorithm performance.

3 Method

In view of the problem that the 3D image space size and voxel size are not uniform in the original data of the brain porcelain, we first resample the image so that the resolution is $1.0 \times 1.0 \times 1.0 \text{ mm}^3$. Then three 2D images in the middle position in the 3D image are selected and resize to a size of 128×128 . According to the pre-processing operation, the input size of the network is 128×128 .

The CapsNet network structure is shown in Fig. 3. It is a shallow network structure consisting of only two convolutional layers and one fully connected layer (i.e., Digital Capsule). The network uses the processed 128×128 2D image as a network input. The image first passes through the Pre-Conv layer without pooling. For the image difference between a patient and a normal person, we designed the image of the traditional convolution layer with the step size of 2, the convolution kernel is 9×9 , the output channel number is 256, and the activation function is ReLU. The local feature pre-fetching, the output of this layer is $60 \times 60 \times 256$, which can reduce the problem of receptive domain overlap. Traditionally, CNN and fully connected networks use scalars as entities, while capsules use multidimensional vectors as entities, which facilitates network feature extraction for entities. Among them, the Pre-Conv layer is the lowest level of multidimensional entities [6]. The Pre-Conv layer output is transferred to the Capsule Layer (shown in the red dashed box in Fig. 2). The upper vector input \mathbf{u}_i is multiplied by \mathbf{W}_{ij} to get the middle vector \mathbf{u}_{ji} . The vector \mathbf{s}_j is the sum of the product of c_{ij} and \mathbf{u}_{ji} . The c_{ij} is updated by *Dynamic Routing* algorithm (see in 3.5) with \mathbf{u}_{ji} and the *Capsule's* output, which is calculated from *Squashing* activation function (see in 3.4). The capsule layer mainly comprises two parts: A main capsule sublayer and a digital capsule sublayer.

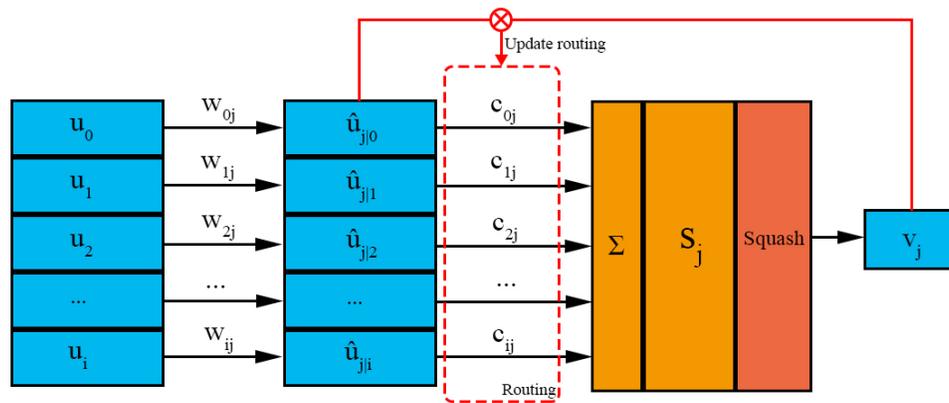


Figure 2: Capsule

3.1 Primary Capsule

The Primary Capsule is the first layer of the capsule layer and consists of an inactive function and a non-pooled convolution and reshape. This layer is intended to convert the non-“encapsulated” Pre-Conv layer feature input of the upper layer into an “encapsulated” feature for later layer processing. For the brain MRI data, we design a step size of 2, a convolution kernel size of 9×9 , and a traditional convolutional layer of $26 \times 26 \times 256$ size to maintain a small spatial dimension. Unlike traditional CNN, this layer does not design a pooling layer or uses an activation function, but instead converts the output to a $26 \times 26 \times 32 \times 8$ capsule layer output. Among them, 32×8 in the output can be regarded as a vector output of 32 channels with a dimension of 8 (8D), or as a 676 (i.e., 26×26) 32-channel 8D “one-dimensional” vector output. And in the 26×26 size region, weight sharing between these capsules reduces the over-fitting problem by reducing the training parameters. The dimensional transformation here is the soul of CapsNet, and it is also the original intention of the capsule design. The processing may include vector information of some special information, such as similarity, direction, size, angle, etc., which is more conducive to improving the classification performance, and finally determines the activation value (output value) of the capsule neuron according to the modulus length of the vector.

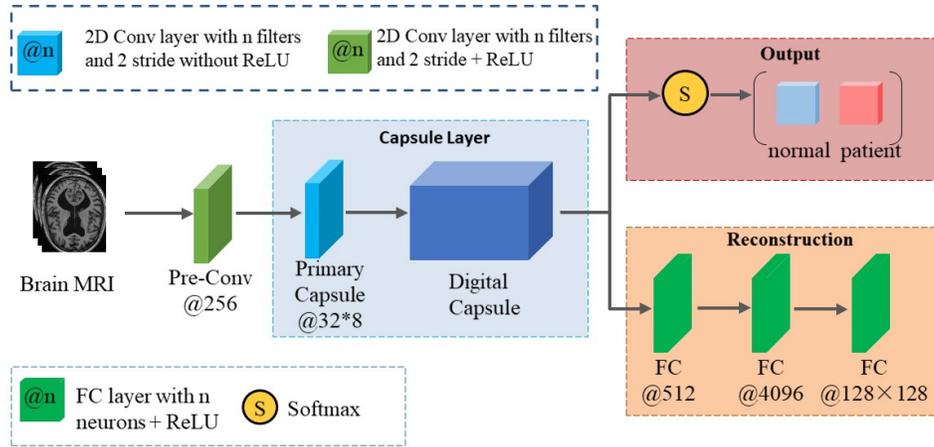


Figure 3: Capsule network

3.2 Digital Capsule

The Digital Capsule is the second layer of the capsule layer. The digital capsule layer uses a vector as input compared to the input in the traditional neuron as a scalar. The main function of this layer is to replace the traditional Fully Connected Layer (FC), transforming the 676×32 8-D capsule of the main capsule sub-layer into two (for the two-class problem) 8-D capsule, and finally using it. The vector modulus length of the two capsules represents the predicted probability of the corresponding two classifications. The digital capsule sublayer is the core of the capsule network. The two main capsule-specific algorithms are applied to this layer. The main model is shown in Fig. 3. The network uses brain MRI as input. Where Pre-Conv layer (light green) is a simple CNN layer for extracting features. Capsule layer is the point of the network including Primary Capsule layer (light-blue) and Digital Capsule layer (blue). The output of the Capsule layer is connected to Output (pink part) and Reconstruction layers (orange part) for inferring or training.

Let the input of the first layer capsule be $K^{(l)}$ $D^{(l)}$ -dimensional vector \mathbf{u}_i ($\mathbf{u}_i \in \mathbb{R}^{(l)}$, $i \in (0, \dots, K^{(l)})$) single capsule concrete realization equation as shown in Eqs. (1)–(4).

$$\hat{\mathbf{u}}_{ij} = \mathbf{u}_i \mathbf{W}_{ij} \quad (1)$$

where $\hat{\mathbf{u}}_{ij} \in \mathbb{R}^{D^{(l+1)}}$, $\mathbf{W}_{ij} \in \mathbb{R}^{D^{(l)} \times D^{(l+1)}}$, $\hat{\mathbf{u}}_{ij}$ is the affine transformation of \mathbf{u}_i , the transformation requires that each row or column in the matrix \mathbf{W} is 1 and \mathbf{W}_{ij} is the weight matrix.

$$\mathbf{c}_i = \text{Softmax}(\mathbf{b}_i) \quad (2)$$

Among them, the weight \mathbf{c}_i is obtained by the “routing softmax” operation, and b_{ij} is the logarithmic prior probability of the capsule i and the capsule j , which is obtained by iterative calculation by the “dynamic routing” algorithm (see 3.5 for details).

$$\mathbf{s}_j = \sum_i c_{ij} \hat{\mathbf{u}}_{j|i} \quad (3)$$

where \mathbf{s}_j is the weighted sum of the “predictive vector” calculated by $\hat{\mathbf{u}}_{j|i}$.

$$\mathbf{v}_j = \text{squashing}(\mathbf{s}_j) \quad (4)$$

where \mathbf{v}_j is \mathbf{s}_j calculated by Eq. (3) (see 3.4 for details). Eq. (3) is similar to the weighted summation operation of traditional neurons: $a_{j|i} = w_{ij} x_i + b$. Eq. (4) is similar to the activation function of traditional neurons such as sigmoid, ReLU, tanh and so on.

3.3 Reconstruction Layer

The reconstruction layer is next to the capsule layer and consists of three fully connected layers. We design dimensions for the brain MRI dataset size of 512, 4096, and $128 \times 128 = 16384$, respectively. In order to introduce the reconstructed error loss function to optimize the model, the output of the last layer of the reconstructed block is designed as the input image size. The introduction of reconstruction errors in this network structure can effectively improve the accuracy of the overall model.

3.4 Squashing

It can be seen from Fig. 2 and Eq. (4) that the capsule activation function is a Squashing function, and the specific calculation Eq. (4) is as shown in the Eq. (5). This function is a non-linear activation function that maintains the input vector dimensions unchanged. This function has the following characteristics:

- The value range of \mathbf{v}_j is limited to $[0,1)$, so the length of the output vector can represent the probability of occurrence of an entity. The larger the modulus, the greater the probability that the entity will appear.
- The function is monotonically increasing, so “encourage” the long-formed vector and “compress” the smaller vector.

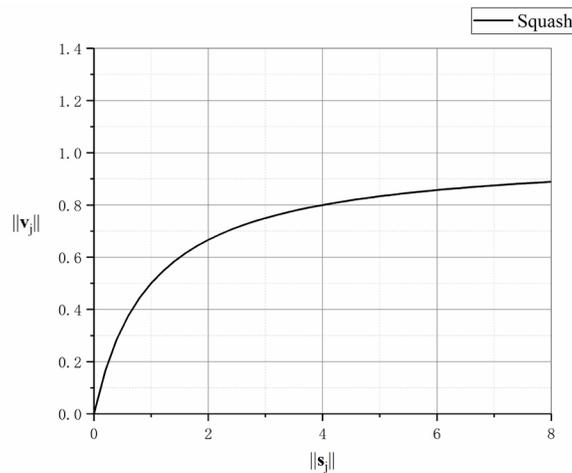


Figure 4: Squashing activation function

$$\mathbf{v}_i = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{5}$$

That is, the “activation function” of Capsule is actually a compression and redistribution of the length of the vector. The relationship between the modulus of the function output vector and the mode of the input vector is shown in Fig. 4. Above shown the function curve of the modulus of the input vector s_j and the modulus of the output vector v_j . The function curve of the Squashing function is similar to the positive semi-axis part of the Sigmoid function. When the input mode length approaches infinity, the output approaches 1.

3.5 Dynamic Routing

From Eq. (2), the capsule network needs to iteratively calculate the logarithmic prior probability of capsules and capsules through a dynamic routing algorithm. Hinton pointed out that “finding the best (processing) path is equivalent to (correctly) processing the image”. This is one of the reasons for introducing dynamic routing in the Capsule framework.

One way to find the “best path” is to pick the input vector that best matches the output vector. The degree of compliance is characterized by the inner product of the output vector and the input vector (the linearly transformed vector). The algorithm gains greater weight by updating the iterative weights to make input vectors that contribute more to the output vector. The specific algorithm is shown in Algorithm 1, and its action area is shown by a red dashed box in FIG. This update algorithm is easy to converge. The literature [6] considers that the algorithm is iterative 3 times. Similar to other algorithms, dynamic routing also has over-fitting problems. Although increasing the number of iterations of dynamic routing can improve the recognition accuracy, it will reduce the generalization of the algorithm.

The \mathbf{c}_i in Eq. (3) is the sum of the logarithmic prior probabilities calculated for the upper input capsule (see the Algorithm 1 Step (3) for its value update), initialized to 0 in the first iteration, and passed the Algorithm 1 Step (6) iterative update. The weight \mathbf{c}_i is calculated by the softmax function in Step (3). Finally, the capsules i are sorted by their prior probability from large to small and passed to the capsule of the next layer. The “prediction vector” \mathbf{s}_j in the Eq. (3) is calculated by the Step (4) of the algorithm 1. Finally, the output $(l+1)$ th layer capsule \mathbf{v}_j is calculated by the Step (5) of Algorithm 1 (see the Eq. (5) for the Squashing function). Finally, the input $\hat{\mathbf{u}}_{j|i}$ is multiplied by the output \mathbf{v}_j and added to b_{ij} , and the value is the updated value of b_{ij} . The dynamic routing algorithm is shown in Algorithm 1.

Algorithm 1 Dynamic Routing

Require: Input vector $\hat{\mathbf{u}}_{j|i}$; route number r ; The number of current layer l

Ensure: Output vector \mathbf{v}_j

1: For all l layer capsules i and all $(l+1)$ layer capsules j : $\mathbf{b} \leftarrow \mathbf{0}$;

2: **while** $r \neq 0$ **do**

3: For all l layer capsules i :

$$\mathbf{c}_i \leftarrow \text{softmax}(\mathbf{b}_i)$$

4: For all $(l+1)$ layer capsules j :

$$\mathbf{s}_j \leftarrow \sum c_{ij} \hat{\mathbf{u}}_{j|i}$$

5: For all $(l+1)$ layer capsules j :

$$\mathbf{v}_j \leftarrow \text{squash}(\mathbf{s}_j)$$

6: For all l layer capsules i and all $(l+1)$ layer capsules j :

$$b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$$

7: $r \leftarrow r - 1$

8: **end while**

9: **return** \mathbf{v}_j

4 Method

We divide the loss function into two parts: Boundary loss and reconstruction loss. In the capsule network, we pass the vector length of a capsule as the probability of the existence of this entity. Boundary loss: Use a loss function similar to SVM.

$$L_T = T_c \max(0, m^+ - \|\mathbf{v}_c\|)^2 \quad (6)$$

$$L_F = \lambda (1 - T_c) \max(0, \|\mathbf{v}_c\| - m_-)^2 \quad (7)$$

$$L_{margin} = L_m + L_r \quad (8)$$

where T_c is the c category information. $T_c = 1$, if the c category exists, else $T_c = 0$. In Eq. (6) and Eq. (7), m^+ and m^- take 0.9 and 0.1, respectively, as the upper and lower boundaries of $\|\mathbf{v}_c\|$ to avoid false positives and false negatives.

5 Experiments

5.1 Experimental Configuration

The experiment was based on Python 3.6.7, Ubuntu 16.04, and Tensorflow 1.10, running on a GPU model NVIDIA GeForce RTX 2080 TI. To ensure the consistency of the results of the comparison experiment (ResNet18, ResNeXt50, DenseNet121, VGG16), we trained 500 cycles (batch size is 32). Each experiment used a five-fold cross-validation to select the test set and training set. Finally, the average value of each index of the results of the five experiments was used as the evaluation amount. The specific settings of each experiment are as follows:

CapsNet: Super parameters μ and λ are set to 0.0005 and 0.5 respectively. The loss function is same as Eq. (8). The trained optimizer uses Adam optimizer [15] and the learning rate is adjusted to 1E-3. The total number of training rounds is 500 rounds [16–18].

ResNet18 ResNeXt50 DensetNet121 VGG16: The loss function is categorical cross-entropy. The trained optimizer uses the Adam optimizer and the learning rate is adjusted to 1E-3. The total number of training rounds is 500 rounds.

5.2 Dataset

The data set is a 3D MRI scan of the human head and consists of three categories, healthy samples, MCI samples, and AD samples. The data set contains 68 samples of Alzheimer's disease patients, 151 samples of MCI patients, and 81 samples of healthy people.

5.2 Analysis of Experiments

The average accuracy and standard deviation of the five-fold cross-validation for each model test are shown in Tab. 1. From the accuracy in the table, it can be seen that CapsNet performs well in the automatic diagnosis of MRI patients in the brain. However, it can be seen from the standard deviation in the table that the robustness of CapsNet remains to be verified. The F1-score of five-folder experiment shown in Tab. 1 illustrate that our method got a great performance.

Table 1: Accuracy and standard deviation of the 5-folder experiment

Model	CapsNet	ResNet18	ResNeXt50	DenseNet121	VGG16
Folder 1	0.8500	0.8500	0.8667	0.8000	0.8333
Folder 2	0.9000	0.8167	0.8167	0.8333	0.8333
Folder 3	0.8667	0.8500	0.8167	0.8333	0.8333
Folder 4	0.8500	0.8500	0.8167	0.8500	0.8333
Folder 5	0.8667	0.8167	0.8500	0.8333	0.8333
Max Acc	0.9000	0.8500	0.8667	0.8500	0.8333
Min Acc	0.8500	0.8167	0.8167	0.8000	0.8333
Mean Acc	0.8667	0.8368	0.8333	0.8300	0.8333
Std	0.0204	0.0182	0.0236	0.0183	0.0000

Table 2: F1-score of the 5-folder experiment

Model	F1 _{score}
CapsNet	0.8333
ResNet18	0.8312
ResNeXt50	0.8190
DenseNet121	0.8222
VGG16	0.8014
max F1 _{score}	0.8333
min F1 _{score}	0.8014
mean F1 _{score}	0.8214

Fig. 5 shows the convergence of the various loss functions in the CapsNet network. It can be seen from Fig. 5 that after the boundary loss rapidly converges, the reconstruction error can play an important role in training. When the reconstruction error training begins to decrease, this also illustrates the imaging identity of the data set. Comparing the total loss with the boundary loss, it can be seen that the main contribution of the training comes from the boundary loss, which is consistent with the design idea.

6 Conclusion

CapsNet is shallow, but exceed the deeper traditional neural network of VGG16 in this classification experiment. This shows that the “capsule” structure has great potential for development as a variant of CNN and traditional neurons. Compared to traditional neurons, the neural input of a capsule is no longer a scalar input, but a vector input.

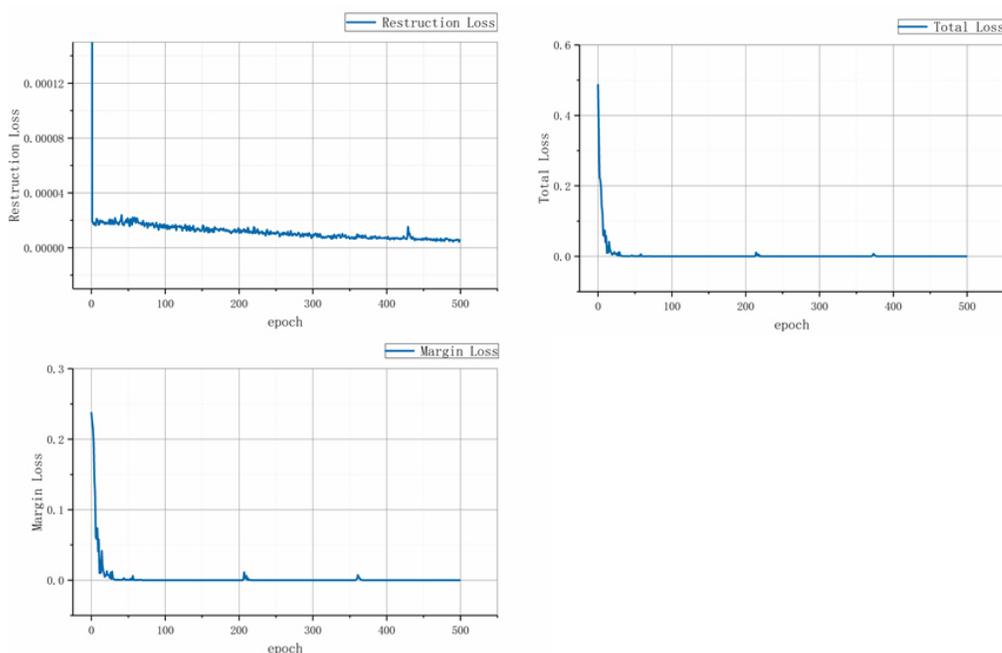


Figure 2: Loss curve. The figure shows reconstruction loss (top left), margin loss (bottom left) and total loss (top right)

CapsNet has a great performance in the data set of this experiment and can perform well in fine-grained classification.

However, it should be pointed out that since the dynamic routing algorithm requires more

dimensional parameters to train, the amount of parameters required by this algorithm increases exponentially with the input and output sizes. Therefore, the structure of CapsNet for a large number of classification problems (such as image segmentation) will consume a lot of GPU display storage resources. We cannot use a single GPU for training, even in the current hardware environment. This also limits the fact that this method is not recommended as an intermediate layer in a multi-layer network, but rather as an output layer (or closer to the output layer).

The main problem facing the brain MRI is that the difference between the diseased image and the normal image is not significant. It is difficult to achieve excellent results using a multi-layer CNN network, and the capsule network CapsNet can be in the case of a shallower network. It accommodates more feature information and is useful for identifying brain MRI. We utilized a shallow capsule network to identify patients with brain MRI in a three-class classification, and compared with VGG16, ResNet18 and so on. The experimental results show that the capsule network is compared with the popular networks in brain MRI recognition. The aspect shows excellent performance.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] K. M. He, X. Y. Zhang, S. Q. Ren and J. Sun, "Deep residual learning for image recognition," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [2] K. Simonyana and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv Preprint*, pp. 1409–1556, 2014.
- [3] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.
- [4] A. Mehrtaash, A. Sedghi, M. Ghafoorian, M. Taghipour, C. M. Tempany *et al.*, "Classification of clinical significance of MRI prostate findings using 3D convolutional neural networks," *Medical Imaging 2017: Computer-Aided Diagnosis*, vol. 10134, 2017.
- [5] J. M. Wolterink, T. Leiner, M. A. Viergeever and I. Išgum, "Automatic segmentation and disease classification using cardiac cine MR images," *Int. Workshop on Statistical Atlases and Computational Models of the Heart*, pp. 101–110, 2017.
- [6] S. Sabour, N. Frosst and G. E. Hinton, "Dynamic routing between capsules," *Advances in Neural Information Processing Systems*, pp. 3856–3866, 2017.
- [7] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated residual transformations for deep neural networks," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017.
- [8] H. Gao, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.
- [9] S. EW. Amin and M. A. Megeed, "Brain tumor diagnosis systems based on artificial neural networks and segmentation using MRI," in *2012 8th Int. Conf. on Informatics and Systems*, MM-119, 2012.
- [10] L. Itti, L. Chang and T. Ernst, "Automatic scan prescription for brain MRI," *Magnetic Resonance in Medicine: An Official Journal of the Int. Society for Magnetic Resonance in Medicine*, vol. 45, no. 3, pp. 486–494, 2001.
- [11] G. Luo, R. An, K. Wang, S. Dong and H. Zhang, "A deep learning network for right ventricle segmentation in short-axis MRI," in *2016 Computing in Cardiology Conf.*, pp. 485–488, 2016.
- [12] G. Luo, G. Sun, K. Wang, S. Dong and H. Zhang. "A novel left ventricular volumes prediction method based on deep learning network in cardiac MRI," in *2016 Computing in Cardiology Conf.*, pp. 89–92, 2016.
- [13] G. E. Hinton, Z. Ghahramani, and Y. W. Teh, "Learning to parse images," *Advances in Neural Information Processing Systems*, pp. 463–469, 2000.

- [14] G. E. Hinton, A. Krizhevsky and S. D. Wang, "Transforming Auto-encoders," in *Int. Conf. on Artificial Neural Networks*, pp. 44–51, 2001.
- [15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Arxiv Preprint*, pp. 1–15, 2014.
- [16] J. Su, G. Wen and D. Hong, "A new RFID anti-collision algorithm based on the Q-ary search scheme," *Chinese Journal of Electronics*, vol. 24, no. 4, pp. 679–683, 2015.
- [17] H. Chen, K. Liu, C. Ma, Y. Han and J. Su, "A novel time-aware frame adjustment strategy for RFID anti-collision," *Computers, Materials & Continua*, vol. 57, no. 2, pp. 195–204, 2018.
- [18] J. Su, X. Zhao, D. Hong, Z. Luo and H. Chen, "Q-value fine-grained adjustment based RFID anti-collision algorithm," *IEICE Transactions on Communications*, vol. E99-B, no. 7, pp. 1593–1598, 2016.